Parameterized coloring problems on chordal graphs*

Dániel Marx

Department of Computer Science and Information Theory,
Budapest University of Technology and Economics
Budapest, H-1521, Hungary
dmarx@cs.bme.hu

Abstract. In the precoloring extension problem (PREXT) a graph is given with some of the vertices having a preassigned color and it has to be decided whether this coloring can be extended to a proper coloring of the graph with the given number of colors. Two parameterized versions of the problem are studied in the paper: either the number of precolored vertices or the number of colors used in the precoloring is restricted to be at most k. We show that these problems are polynomial time solvable but W[1]-hard in chordal graphs. For a graph class $\mathscr{F}$, let $\mathscr{F} + ke$ (resp. $\mathscr{F} + kv$) denote those graphs that can be made to be a member of $\mathscr{F}$ by deleting at most k edges (resp. vertices). We investigate the connection between PREXT in $\mathscr{F}$ and the coloring of $\mathscr{F} + ke$, $\mathscr{F} + ve$ graphs. Answering an open question of Leizhen Cai [5], we show that coloring chordal+ke graphs is fixed-parameter tractable.

1 Introduction

In graph vertex coloring we have to assign colors to the vertices such that neighboring vertices receive different colors. In the *precoloring extension* (PREXT) problem a subset W of the vertices have a preassigned color and we have to extend this to a proper k-coloring of the whole graph. Since vertex coloring is the special case when $W = \emptyset$, the precoloring extension problem is NP-complete in every class of graphs where vertex coloring is NP-complete. See [2,7,8] for more background and results on PREXT.

In this paper we study the precoloring extension problem on chordal graphs. PREXT is NP-complete for interval graphs [2] (and for unit interval graphs [12]), hence it is NP-complete for chordal graphs as well. On the other hand, if every color is used only once in the precoloring (this special case is called 1-PREXT), then the problem becomes polynomial time solvable for interval graphs [2], and more generally, for chordal graphs [11]. Here we introduce two new restricted versions of PREXT: we investigate the complexity of the problem when either there are only k precolored vertices, or there are only k colors used in the precoloring.

* Research is supported in part by grants OTKA 44733, 42559 and 42706 of the Hungarian National Science Fund.

Clearly, the former is a special case of the latter. By giving an $O(kn^{k+2})$ time algorithm, we show that for fixed k both problems are polynomial time solvable on chordal graphs. However, we cannot expect to find a uniformly polynomial time algorithm for these problems, since they are W[1]-hard even for interval graphs. To establish W[1]-hardness, we use the recent result of Slivkins [15] that the edge-disjoint paths problem is W[1]-hard.

Leizhen Cai [5] introduced a whole new family of parameterized problems. If $\mathscr{F}$ is an arbitrary class of graphs, then denote by $\mathscr{F} - kv$ (resp. $\mathscr{F} - ke$) the class of those graphs that can be obtained from a member of $\mathscr{F}$ by deleting at most k vertices (resp. k edges). Similarly, let $\mathscr{F} + kv$ (resp. $\mathscr{F} + ke$) be the class of those graphs that can be made to be a member of $\mathscr{F}$ by deleting at most k vertices (resp. k edges). For any class of graphs $\mathscr{F}$ and for any graph problem, we can ask what is the complexity of the problem restricted to these 'almost $\mathscr{F}$' graphs. This question is investigated in [5] for the vertex coloring problem. Coloring $\mathscr{F} + kv$ or $\mathscr{F} + ke$ graphs can be very different than coloring graphs in $\mathscr{F}$, and might involve significantly new approaches.

We investigate the relations between PrExt and the coloring of the modified graph classes. We show that for several reasonable graph classes, reductions are possible between PrExt for graphs in $\mathscr{F}$ and the coloring of $\mathscr{F} + kv$ or $\mathscr{F} + ke$ graphs. Based on this correspondence between the problems, we show that both chordal+ke and chordal+kv graphs can be colored in polynomial time for fixed k, but chordal+kv graph coloring is W[1]-hard. Moreover, answering an open question of Cai [5], we develop a uniformly polynomial time algorithm for coloring chordal+ke graphs.

The paper is organized as follows. Section 2 contains preliminary notions. Section 3 reviews tree decomposition, which will be our main tool when dealing with chordal graphs. In Section 4, we investigate the parameterized PrExt problems for chordal graphs. The connections between PrExt and coloring $\mathscr{F} + ke$, $\mathscr{F} + kv$ graphs are investigated in Section 5. Finally, in Section 6, we show that coloring chordal+ke graphs is fixed-parameter tractable.

2 Preliminaries

A C-coloring is a proper coloring of the vertices with color set C. We introduce two different parameterization of the precoloring extension problem. Formally, the problem is as follows:

> **Precoloring Extension (PrExt)**
> *Input:* A graph $G(V, E)$, a set of colors C, and a precoloring $\psi: W \to C$ for a set of vertices $W \subseteq V$.
> *Parameter 1:* $|W|$, the number of precolored vertices.
> *Parameter 2:* $|\{\psi(w) : w \in W\}| = |C_W|$, the number of colors used in the precoloring.
> *Question:* Is there a proper C-coloring ψ' of G that extends ψ (i.e., $\psi'(w) = \psi(w)$ for every $w \in W$)?

Note that $C_W \subseteq C$ is the set of colors appearing on the precolored vertices, and can be much smaller than the set of available colors C. When we consider parameter 1, then the problem will be called PrExt with fixed number of precolored vertices, while considering parameter 2 corresponds to PrExt with fixed number of colors in the precoloring.

For every class $\mathscr{F}$ and every fixed k, one can ask what is the complexity of vertex coloring on the four classes $\mathscr{F} + ke$, $\mathscr{F} + kv$, $\mathscr{F} - ke$, $\mathscr{F} - kv$. The first question is whether the problem is NP-complete for some fixed k. If the problem is solvable in polynomial time for every fixed k, then the next question is whether the problem is fixed-parameter tractable, that is, whether there is a uniformly polynomial time algorithm for the given classes.

If $\mathscr{F}$ is hereditary with respect to taking induced subgraphs, then $\mathscr{F} - kv$ is the same as $\mathscr{F}$, hence coloring $\mathscr{F} - kv$ graphs is the same as coloring in $\mathscr{F}$. Moreover, it is shown in [5] that if $\mathscr{F}$ is closed under edge contraction and has a polynomial time algorithm for coloring, then coloring $\mathscr{F} - ke$ graphs is fixed parameter tractable. Therefore we can conclude that coloring chordal$-kv$ and chordal$-ke$ graphs are in FPT. In this paper we show that coloring chordal$+ke$ graphs is in FPT, but coloring chordal$+kv$ graphs is W[1]-hard.

The *modulator* of an $\mathscr{F} + ke$ graph G is a set of at most k edges whose removal makes G a member of $\mathscr{F}$. Similar definitions apply for the other classes. We will call the vertices and edges of the modulator *special edges and vertices*. In the case of $\mathscr{F} + e$ and $\mathscr{F} - e$ graphs, the vertices incident to the special edges are the special vertices.

When considering the complexity of coloring in a given parameterized class, then we can assume either that only the graph is given in the input, or that a modulator is also given. In the case of coloring chordal$-ke$ graphs, this makes no difference as finding the modulator of such a graph (i.e., the at most k edges that can make the graph chordal) is in FPT [4,9]. On the other hand, the parameterized complexity of finding the modulator of a chordal$+ke$ graph is open. Thus in our algorithm for coloring chordal$+ke$ graphs, we assume that the modulator is given in the input.

3 Tree decomposition

A graph is *chordal* if it does not contain a cycle of length greater than 3 as an induced subgraph. This section summarizes some well-known properties of chordal graphs. First, chordal graphs can be also characterized as the intersection graphs of subtrees of a tree (see e.g. [6]):

Theorem 1. *The following two statements are equivalent:*

1. *$G(V, E)$ is chordal.*
2. *There exists a tree $T(U, F)$ and a subtree $T_v \subseteq T$ for each $v \in V$ such that $u, v \in V$ are neighbors in $G(V, E)$ if and only if $T_u \cap T_v \neq \emptyset$.*

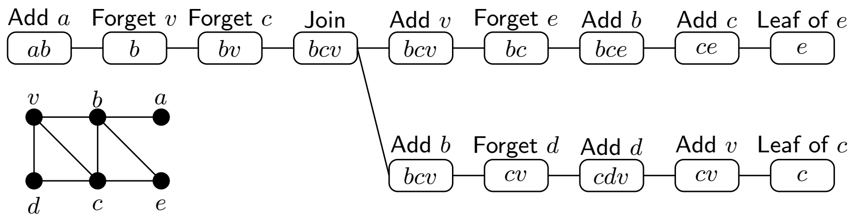

Fig. 1. Nice tree decomposition of a chordal graph.

The tree T together with the subtrees T_v is called the *tree decomposition* of G. A tree decomposition of G can be found in polynomial time (see [6,14]).

We assume that T is a rooted tree with some root $r \in U$. For clarity, we will use the word 'vertex' when we refer to the graph $G(V, E)$, and 'node' when referring to $T(U, F)$. For a node $x \in U$, denote by V_x those vertices whose subtree contains x or a descendant of x. The subgraph of G induced by V_x will be denoted by $G_x = G[V_x]$. For a node $x \in U$ of T, denote by K_x the union of v's where $x \in V(T_v)$. Clearly, the vertices of K_x are in V_x, and they form a clique in G_x, since the corresponding trees intersect in T at node x. An important property of the tree decomposition is the following: for every node $x \in U$, the clique K_x separates $V_x \setminus K_x$ and $V \setminus V_x$. That is, among the vertices of V_x, only the vertices in K_x can be adjacent to $V \setminus V_x$.

A tree decomposition will be called *nice* [10], if it satisfies the following additional requirements (see Figure 1):

- Every node $x \in U$ has at most two children.
- If $x \in U$ has two children $y, z \in U$, then $K_x = K_y = K_z$ (x is a *join* node).
- If $x \in U$ has only one child $y \in U$, then either $K_x = K_y \cup \{v\}$ (x is an *add* node) or $K_x = K_y \setminus \{v\}$ (x is a *forget* node) for some $v \in V$.
- If $x \in U$ has no children, then K_x contains exactly one vertex (x is a *leaf* node).

By splitting the nodes of the tree in an appropriate way, a tree decomposition of G can be transformed into a nice tree decomposition in polynomial time.

A vertex v can have multiple add nodes, but at most one forget node (the vertices in clique K_r of the root r have no forget nodes, but every other vertex has exactly one). For a vertex v, its subtree T_v is the subtree rooted at the forget node of v (if it exists, otherwise at the root) and whose leaves are exactly the add nodes and leaf nodes of v.

4 PrExt on chordal graphs

In this section we show that PrExt can be solved in polynomial time for chordal graphs if the number of colors used in the precoloring is bounded by a constant k. The algorithm presented below is a straightforward application of the tree decomposition described in Section 3. The running time of the algorithm is $O(kn^{k+2})$, hence it is not uniformly polynomial. However, in Theorem 3 it is shown that the problem is W[1]-hard, hence we cannot hope to find a uniformly polynomial algorithm.

Theorem 2. *The* PREXT *problem can be solved in* $O(kn^{k+2})$ *time for chordal graphs, if the number of colors in the precoloring is at most* k.

Proof. It can be assumed that the colors used in the precoloring are the colors 1, 2, ..., k. For each node x of the nice tree decomposition of the graph, we solve several subproblems using dynamic programming. Each subproblem is described by a vector $[\alpha_1, \dots, \alpha_k]$, where each α_i is either a vertex of K_x, or the symbol $\star$. We say that such a vector is *feasible* for node x, if there is a precoloring extension for G_x with the following property: if α_i $(1 \le i \le k)$ is $\star$, then color i does not appear on the clique K_x, otherwise it appears on vertex $\alpha_i \in K_x$. Notice that in a feasible vector a vertex can appear at most once (but the star can appear several times), thus in the following we consider only such vectors.

Clearly, the precoloring can be extended to the whole graph if and only if the the root node r has at least one feasible vector. The algorithm finds the feasible vectors for each node of T. We construct the feasible vectors for the nodes in a bottom-up fashion. First, they are easy to determine for the leaves. Moreover, they can be constructed for an arbitrary node if the feasible vectors for the children are already available. The techniques are standard, details omitted. $\square$

To prove that PREXT with fixed number of precolored vertices is W[1]-hard for interval graphs, we use reduction from the edge disjoint paths problem, which is the following:

Edge disjoint paths
Input: A directed graph $G(V, E)$, with k pairs of vertices (s_i, t_i).
Parameter: The number of pairs k.
Question: Is there a set of k pairwise edge disjoint directed paths P_1, ..., P_k such that path P_i goes from s_i to t_i?

Recently, Slivkins [15] proved that the edge disjoint paths problem is W[1]-hard for directed acyclic graphs.

Theorem 3. PREXT *with fixed number of precolored vertices is* W[1]*-hard for interval graphs.*

Proof. The proof is by a parameterized reduction from the directed acyclic edge disjoint path problem. Given a directed acyclic graph $G(V, E)$ and terminal pairs s_i, t_i $(1 \le i \le k)$, we construct an interval graph with $k' = 2k$ precolored vertices in such a way that the interval graph has a precoloring extension if and only if the disjoint paths problem can be solved. Let 1, 2, ..., n be the vertices of G in a topological ordering. For each edge $\overrightarrow{xy}$ of G we add an interval $[x, y)$. For each terminal pair s_i, t_i we add two intervals $[0, s_i)$ and $[t_i, n+1)$, and precolor these intervals with color i.

Denote by $\ell(x)$ the number of intervals whose *right* end point is x (i.e., the intervals that arrive to x from the left), and by $r(x)$ the number of intervals whose left end point is x. In other words, $\ell(x)$ is the number of edges entering x plus the number of demands starting in x. If $\ell(x) < r(x)$, then add $r(x) - \ell(x)$ new intervals $[0, x)$ to the graph, if $\ell(x) > r(x)$, then add $\ell(x) - r(x)$ new intervals $[x, n+1)$. A consequence of this is that each point of $[0, n+1)$ is contained in the same number (denote it by c) of intervals: at each point the number of intervals ending equals the number of intervals starting. We claim that the interval graph has a precoloring extension with c colors if and only if the disjoint paths problem has a solution.

Assume first that there are k disjoint paths joining the terminal pairs. For each edge $\overrightarrow{xy}$, if it is used by the ith terminal pair, then color the interval $[x, y)$ with color i. Notice that the intervals we colored with color i do not intersect each other, and their union is exactly $[s_i, t_i)$. Therefore, considering also the two intervals $[0, s_i)$ and $[s_i, n+1)$ precolored with color i, each point of $[0, n+1)$ is covered by exactly one interval with color i. Therefore each point is contained in exactly $c - k$ intervals that do not have a color yet. This means that the uncolored intervals induce an interval graph where every point is in exactly $c-k$ intervals, and it is well-known that such an interval graph has clique number $c - k$ and can be colored with $c - k$ colors. Therefore the precoloring can be extended using $c - k$ colors in addition to the k colors used in the precoloring.

Now assume that the precoloring can be extended using c colors. Since each point in the interval $[0, n+1)$ is covered by exactly c intervals, therefore each point is covered by an interval of color i. Thus if an interval with color i ends at point x, then an interval with color i has to start at x. Since the interval $[0, s_i)$ has color i, there has to be an interval $[s_i, s_{i,1})$ with color i. Similarly, there has to be an interval $[s_{i,1}, s_{i,2})$ with color i, etc. Continuing this way, we will eventually arrive to an interval $[s_{i,p}, t_i)$. By the way the intervals were constructed, the edges $\overrightarrow{s_i s_{i,1}}$, $\overrightarrow{s_{i,1} s_{i,2}}$, $\ldots$, $\overrightarrow{s_{i,p} t_i}$ form a path from s_i to t_i. It is clear that the paths for different values of i are disjoint since each interval has only one color. Thus we constructed a solution to the disjoint paths problem, as required. $\square$

5 Reductions

In this section we give reductions between PrExt on $\mathscr{F}$ and coloring $\mathscr{F} + kv$, $\mathscr{F} + ke$ graphs. It turns out that if $\mathscr{F}$ is closed under disjoint union and attaching pendant vertices, then

$$\begin{aligned} &\text{coloring } \mathscr{F} + ke \text{ graphs} \preceq \text{PrExt on } \mathscr{F} \text{ with fixed } |W| \preceq \\ &\text{coloring } \mathscr{F} + kv \text{ graphs} \preceq \text{PrExt on } \mathscr{F} \text{ with fixed } |C_W| \end{aligned}$$

When coloring $\mathscr{F} + ke$ or $\mathscr{F} + kv$ graphs, we assume that the modulator of the graph is given in the input. The proof of the following four results will appear in the full version:

Theorem 4. *For every class $\mathscr{F}$ of graphs, coloring $\mathscr{F}+ke$ graphs can be reduced to* PrExt *with fixed number of precolored vertices, if the modulator of the graph is given in the input.*

Theorem 5. *Let $\mathscr{F}$ be a class of graphs closed under attaching pendant vertices. Coloring $\mathscr{F} + kv$ graphs can be reduced to* PrExt *with fixed number of colors in the precoloring, if the modulator of the graph is given in the input.*

Theorem 6. *If $\mathscr{F}$ is a hereditary graph class closed under disjoint union, then* PrExt *in $\mathscr{F}$ with fixed number of precolored vertices can be reduced to the coloring of $\mathscr{F} + kv$ graphs.*

Theorem 7. *If $\mathscr{F}$ is a hereditary graph class closed under joining graphs at a vertex, then* PrExt *on $\mathscr{F}$ with a fixed number of colors in the precoloring can be reduced to the coloring of $\mathscr{F} + kv$ graphs.*

When reducing the coloring of $\mathscr{F} + ke$ or $\mathscr{F} + kv$ graphs to PrExt, the idea is to consider each possible coloring of the special vertices and solve each possibility as a PrExt problem. In the other direction, we use the special edges and vertices to build gadgets that force the precolored vertices to the required colors.

Concerning chordal graphs, putting together Theorems 2–6 gives

Corollary 1. *Coloring chordal+ke and chordal+kv graphs can be done in polynomial time for fixed k. However, coloring interval+kv (hence chordal+kv) graphs is* W[1]*-hard.* □

In Section 6, we improve on this result by showing that coloring chordal+ke graphs is fixed-parameter tractable.

6 Coloring chordal+ke graphs

In Section 5 we have seen that coloring a chordal+ke graph can be reduced to the solution of PrExt problems on a chordal graph, and by Theorem 2, each such problem can be solved in polynomial time. Therefore chordal+ke graphs can be colored in polynomial time for fixed k, but with this algorithm the exponent of n in the running time depends on k. In this section we prove that coloring chordal+ke graphs is fixed-parameter tractable by presenting a uniformly polynomial time algorithm for the problem.

Let H be a chordal+ke graph, and denote by G the chordal graph obtained by deleting the special edges of G. We proceed similarly as in Theorem 2. First we construct a nice tree decomposition of H. A subgraph G_x of G corresponds to each node x of the nice tree decomposition. Let H_x be the graph G_x plus the special edges induced by the vertex set of G_x. For each subgraph H_x, we try to find a proper coloring. In fact, for every node x we solve several subproblems: each subproblem corresponds to finding a coloring of H_x with a given property (to be defined later). The main idea of the algorithm is that the number of subproblems considered at a node can be reduced to a function of k.

Before presenting the algorithm, we introduce a technical tool that will be useful. For each node x of the nice tree decomposition, the graph H_x^* is defined by adding a clique of $|C|-|K_x|$ vertices $u_1, u_2, \ldots, u_{|C|-|K_x|}$ to the graph H_x, and connecting each new vertex to each vertex of K_x. The clique K_x together with the new vertices form a clique of size $|C|$, this clique will be called K_x^*. Instead of the colorings of H_x, we will consider the colorings of H_x^*. Although H_x^* is a supergraph of H_x, it is not more difficult to color than H_x: the new vertices are only connected to K_x, hence in every coloring of H_x there remains $|C|-|K_x|$ colors from C to color these vertices. However, considering the colorings of H_x^* instead of the colorings of H_x will make the arguments cleaner. The reason for this is that in every C-coloring of H_x^* every color of C appears on the clique K_x^* exactly once, which makes the description of the colorings more uniform.

Another technical trick is that we will assume that every special vertex is contained in exactly one special edge (recall that a vertex is called special if it is the end point of a special edge.) A graph can be transformed to such a form without changing the chromatic number, details omitted. The idea is to replace a special vertex with multiple vertices, and add some simple gadgets that force these vertices to have the same color. Since each special vertex is contained in only one special edge, thus each special vertex w has a unique *pair*, which is the other vertex of the special edge incident to w.

Now we define the subproblems associated with node x. A set system is defined where each set corresponds to a type of coloring that is possible on H_x^*. Let W be the set of special vertices, we have $|W| \leq 2k$. Let W_x be the special vertices contained in the subgraph H_x^*. In the following, we consider sets over $K_x^* \times W$. That is, each element of the set is a pair (v, w) with $v \in K_x^*$, $w \in W$.

Definition 1. *To each C-coloring ψ of H_x^*, we associate a set $S_x(\psi) \subseteq K_x^* \times W$ such that $(v, w) \in S_x(\psi)$ ($v \in K_x^*$, $w \in W_x$) if and only if $\psi(v) = \psi(w)$. The set system $\mathscr{S}_x$ over $K_x^* \times W$ contains a set S if and only if there is a coloring ψ of H_x^* such that $S = S_x(\psi)$.*

The set $S_x(\psi)$ describes ψ on H_x^* as it is seen from the "outside", i.e., from $H \setminus H_x^*$. In H_x^* only K_x^* and W_x are connected to the outside. Since K_x^* is a clique of size $|C|$, every color appears on exactly one vertex, this is the same for every coloring. Seen from the outside, the only difference between the colorings is how the colors are assigned to W_x. The set $S_x(\psi)$ captures this information.

Subgraph H_x^* (hence H_x) is C-colorable if and only if the set system $\mathscr{S}_x$ is not empty. Therefore to decide the C-colorability of H, we have to check whether $\mathscr{S}_r$ is empty, where r is the root of the nice tree decomposition.

Before proceeding further, we need some new definitions.

Definition 2. *A set $S \subseteq K_x^* \times W$ is* regular*, if for every $w \in W$, there is at most one element of the form (v, w) in S. Moreover, we also require that if $v \in K_x^* \cap W$ then $(v, v) \in S$. The set S* contains *vertex w, if there is an element (v, w) in S for some $v \in K_x^*$.*

For a coloring ψ of H_x^*, set $S_x(\psi)$ is regular and contains only vertices from W_x.

Definition 3. *For a set* $S \in K_x^* \times W$*, its* blocker $B(S)$ *is a subset of* $K_x^* \times W$ *such that* $(v,w) \in B(S)$ *if and only if* $(v,w') \in S$ *for the pair* w' *of* w*. We say that sets* S_1 *and* S_2 *form a* non-blocking pair *if* $B(S_1) \cap S_2 = \emptyset$ *and* $S_1 \cap B(S_2) = \emptyset$*.*

If ψ is a coloring of H_x^*, then the set $B(S_x(\psi))$ describes the requirements that have to be satisfied if we want to extend ψ to the whole graph. For example, if $(v,w) \in S_x(\psi)$, then this means that $v \in K_x^*$ has the same color as special vertex w. Now $(v,w') \in B(S_x(\psi))$ for the pair w' of w. This tells us that we *should not* color w' with the same color as v, because in this case the pairs w and w' would have the same color.

To be a non-blocking pair, it is sufficient that one of $B(S_1)\cap S_2$ and $S_1 \cap B(S_2)$ is empty:

Lemma 1. *For two sets* $S_1, S_2 \in K_x \times W$*, we have that* $B(S_1) \cap S_2 = \emptyset$ *if and only if* $S_1 \cap B(S_2) = \emptyset$*.*

Proof. Suppose that $B(S_1)\cap S_2 = \emptyset$, but $(v,w) \in S_1 \cap B(S_2)$ (the other direction follows by symmetry). Since $(v,w) \in B(S_2)$, this means that $(v,w') \in S_2$ where w' is the pair of w. But in this case $(v,w) \in S_1$ implies that $(v,w') \in B(S_1)$, contradicting $B(S_1) \cap S_2 = \emptyset$. □

The following lemma motivates the definition of the non-blocking pair, it turns out to be very relevant to our problem. If x is a join node, then we can give a new characterization of $\mathscr{S}_x$, based on the set systems of its children.

Lemma 2. *If* x *is a join node with children* y *and* z*, then*

$$\mathscr{S}_x = \{S_y \cup S_z : S_y \in \mathscr{S}_y \text{ and } S_z \in \mathscr{S}_z \text{ form a non-blocking pair}\}.$$

Proof. If $S \in \mathscr{S}_x$, then there is a corresponding coloring ψ of H_x^*. Coloring ψ induces a coloring ψ_y (resp. ψ_z) of H_y^* (resp. H_z^*). Let S_y (resp. S_z) be the set that corresponds to coloring ψ_y (resp. ψ_z). We show that S_y and S_z form a non-blocking pair, and $S = S_y \cup S_z$. By Lemma 1, it is enough to show that $S_y \cap B(S_z) = \emptyset$. Suppose that $S_y \cap B(S_z)$ contains the element (v,w) for some $v \in K_y^* = K_z^*$ and $w \in W_y$. By the definition of S_y, this means that $\psi_y(v) = \psi_y(w)$. Since $(v,w) \in B(S_z)$, thus $(v,w') \in S_z$ for the pair $w' \in W$ of w. Therefore $\psi_z(v) = \psi_z(w')$ follows. However, $\psi_y(v) = \psi_z(v)$, hence $\psi_y(w) = \psi_z(w')$, which is a contradiction, since w and w' are neighbors, and ψ is a proper coloring of H_x^*. Now we show that $S = S_y \cup S_z$. It is clear that $(v,w) \in S_y$ implies $(v,w) \in S$, hence $S_y \cup S_z \subseteq S$. Moreover, suppose that $(v,w) \in S$. Without loss of generality, it can be assumed that w is contained in H_y^*. This implies that $(v,w) \in S_y$, as required.

Now let $S_y \in \mathscr{S}_y$ and $S_z \in \mathscr{S}_z$ be a non-blocking pair, it has to be shown that $S = S_y \cup S_z$ is in $\mathscr{S}_x$. Let ψ_y (resp. ψ_z) be the coloring corresponding to S_y (resp. S_z). In general, ψ_y and ψ_z might assign different colors to the vertices of $K_x^* = K_y^* = K_z^*$. However, since K_x^* is a clique and every color appears exactly once on it, by permuting the colors in ψ_y, we can ensure that ψ_y and ψ_z agree on K_x^*. We claim that if we merge ψ_y and ψ_z, then the resulting coloring ψ is a proper coloring of H_x^*. The only thing that has to be verified is whether ψ assigns different colors to the end vertices of these special edges that are contained completely neither in H_y^* nor H_z^*. Suppose that special vertices $w \in W_y \setminus W_z$ and $w' \in W_z \setminus W_y$ are pairs, but $\psi(w) = \psi(w')$. We know that $(v, w) \in S_y$ for some $v \in K_y^*$, and similarly $(v', w') \in S_z$. By definition, this means that $\psi_y(v) = \psi_y(w)$ and $\psi_z(v') = \psi(w')$. Since ψ_y and ψ_z assign the same colors to the vertices of the clique K_x^*, thus this is only possible if $v = v'$, implying $(v, w') \in S_z$. However, $B(S_y)$ also contains (v, w') contradicting the assumption that $B(S_y) \cap S_z = \emptyset$. Now it is straightforward to verify that the set corresponding to ψ is $S = S_y \cup S_z$, proving that $S \in \mathscr{S}_x$. □

Lemma 2 gives us a way to obtain the system $\mathscr{S}_x$ if x is a join node and the systems for the children are known. It can be shown for add nodes and forget nodes as well that their set systems can be constructed if the set systems are given for their children. However, we do not prove this here, since this observation does not lead to a uniformly polynomial algorithm. The problem is that the size of $\mathscr{S}_x$ can be $O(n^k)$, therefore it cannot be represented explicitly. On the other hand, in the following we show that it is not necessary to represent the whole set system, most of the sets can be thrown away, it is enough to retain only a constant number of sets.

We will replace $\mathscr{S}_x$ by a system $\mathscr{S}_x^*$ *representative for* $\mathscr{S}_x$ that has constant size. Representative systems and their use in finding disjoint sets were introduced by Monien [13] (and subsequently used also in [1]). Here we give a definition adapted to our problem:

Definition 4. *A subsystem $\mathscr{S}_x^* \subseteq \mathscr{S}_x$ is* representative for $\mathscr{S}_x$ *if the following holds: for each regular set $U \subseteq K_x \times W$ that does not contain vertices in $W_x \setminus K_x^*$, if $\mathscr{S}_x$ contains a set S disjoint from $B(U)$, then $\mathscr{S}_x^*$ also contains a set S' also disjoint from $B(U)$. We say that the subsystem $\mathscr{S}_x^*$ is* minimally representative for $\mathscr{S}_x$*, if it is representative for $\mathscr{S}_x$, but it is not representative after deleting any of the sets from $\mathscr{S}_x^*$.*

That is, if $\mathscr{S}_x$ can present a member avoiding all the forbidden colorings described by $B(U)$, then $\mathscr{S}_x^*$ can present such a member as well. For technical reasons, we are interested only in requirements $B(U)$ with U as described above.

The crucial idea is that the size of a minimally representative system can be bounded by a function of k independent of n (if the size of each set in $\mathscr{S}_x$ is at most $2k$). This is a consequence of the following version of Bollobás' inequality:

Theorem 8 (Bollobás [3]). *Let (A_1, B_1), (A_2, B_2), …, (A_m, B_m) be a sequence of pairs of sets over a common ground set X such that $A_i \cap B_j = \emptyset$ if and only if $i = j$. Then*

$$\sum_{i=1}^{m} \binom{|A_i| + |B_i|}{|A_i|}^{-1} \leq 1.$$

Lemma 3. *If $\mathscr{S}_x^*$ is minimally representative for $\mathscr{S}_x$, then $|\mathscr{S}_x^*| \le \binom{4k}{2k}$.*

Proof. Let $\mathscr{S}_x^* = \{A_1, A_2, \dots, A_m\}$. Since $\mathscr{S}_x^*$ is minimally representative for $\mathscr{S}_x$, therefore for every $1 \le i \le m$, there is a regular set $B_i = B(U_i) \subseteq K_x \times W$ satisfying Definition 4 such that $\mathscr{S}_x$ has a set disjoint from B_i, but A_i is the only set in $\mathscr{S}_x^*$ disjoint from B_i (otherwise A_i could be safely removed from $\mathscr{S}_x^*$). This means that $A_i \cap B_i = \emptyset$, and $A_j \cap B_i \ne \emptyset$ for every $i \ne j$. Therefore (A_1, B_1), (A_2, B_2), ..., (A_m, B_m) satisfy the requirements of Theorem 8, hence

$$1 \ge \sum_{i=1}^{m} \binom{|A_i| + |B_i|}{|A_i|}^{-1} \ge \sum_{i=1}^{m} \binom{|W_x| + |W|}{|W_x|}^{-1} \ge m \binom{4k}{2k}^{-1}.$$

Therefore $m \le \binom{4k}{2k}$, and the lemma follows. □

Lemma 3 shows that one can obtain a constant size representative system by throwing away sets until the system becomes a minimally representative. Another way of obtaining a constant size system is to use the data structure of Monien [13] for finding and storing representative systems. Using that method, we can obtain a representative system of size at most $2k^{2k}$. This can be somewhat larger than $\binom{4k}{2k}$ given by Lemma 3, but it is also good for our purposes.

We show that instead of determining the set system $\mathscr{S}_x$ for each node, it is sufficient to find a set system $\mathscr{S}_x^*$ representative for $\mathscr{S}_x$. That is, if for each child y of x we are given a system $\mathscr{S}_y^*$ representative for $\mathscr{S}_y$, then we can construct a system $\mathscr{S}_x^*$ representative for $\mathscr{S}_x$. For a join node x, one can find a set system $\mathscr{S}_x^*$ representative for $\mathscr{S}_x$ by a characterization analogous to Lemma 2:

Lemma 4. *Let x be a join node with children y and z, and let $\mathscr{S}_y^*$ be representative for $\mathscr{S}_y$, and $\mathscr{S}_z^*$ representative for $\mathscr{S}_z$. Then the system*

$$\mathscr{S}_x^* = \{S_y \cup S_z : S_y \in \mathscr{S}_y^* \text{ and } S_z \in \mathscr{S}_z^* \text{ form a non-blocking pair}\}$$

is representative for $\mathscr{S}_x$.

Proof. Since $\mathscr{S}_y^* \subseteq \mathscr{S}_y$ and $\mathscr{S}_z^* \subseteq \mathscr{S}_z$, by Lemma 2 it follows that $\mathscr{S}_x^* \subseteq \mathscr{S}_x$. Therefore we have to show that for every regular set U not containing vertices from $W_x \setminus K_x^*$, if there is a set $S \in \mathscr{S}_x$ disjoint from $B(U)$, then there is a set $S' \in \mathscr{S}_x^*$ also disjoint from $B(U)$. Let ψ be the coloring corresponding to set S, and let ψ_y (resp. ψ_z) be the coloring of H_y^* (resp. H_z^*) induced by ψ. Let $S_y \in \mathscr{S}_y$ and $S_z \in \mathscr{S}_z$ be the sets corresponding to ψ_y and ψ_z. We have seen in the proof of Lemma 2 that S_y and S_z form a non-blocking pair and $S = S_y \cup S_z$, hence S_y is disjoint from $B(U) \cup B(S_z) = B(U \cup S_z)$. Note that U does not contain vertices from $W_x \setminus K_x^*$, and S_z contains only vertices from H_z^*, hence $U \cup S_z$ is regular, and does not contain vertices from $W_y \setminus K_y^*$. Since $\mathscr{S}_y^*$ is representative for $\mathscr{S}_y$, there is a set $S_y' \in \mathscr{S}_y^*$ that is also disjoint from $B(U \cup S_z)$. By Lemma 1, $S_y' \cap B(S_z) = \emptyset$ implies that $B(S_y') \cap S_z = \emptyset$, hence S_z is disjoint from $U \cup B(S_y') = B(U \cup S_y')$. Since $\mathscr{S}_z^*$ is representative for $\mathscr{S}_z$, there is a set $S_z' \in \mathscr{S}_z^*$ that is also disjoint from $B(U \cup S_y')$. Applying again Lemma 1, we get that S_y' and S_z' form a non-blocking pair, hence $S' = S_y' \cup S_z'$ is in S_x^*. Since S' is disjoint from $B(U)$, thus $\mathscr{S}_x^*$ contains a set disjoint from $B(U)$. □

If x is an add node or forget node with children y and a system $\mathscr{S}_y^*$ representative for S_y is given, then we can construct a system $\mathscr{S}_x^*$ that is representative for $\mathscr{S}_x$. The construction is conceptually not difficult, but requires a tedious discussion. We omit the details.

Therefore starting from the leaves, the systems $\mathscr{S}_x^*$ can be constructed using bottom up dynamic programming. After constructing $\mathscr{S}_x^*$, we use the data structure of Monien to reduce the size of $\mathscr{S}_x^*$ to a constant. This will ensure that each step of the algorithm can be done in uniformly polynomial time. By checking whether $\mathscr{S}_r^*$ is empty for the root r, we can determine whether the graph has a C-coloring. This proves the main result of the section:

Theorem 9. *Coloring chordal+ke graphs is in* FPT *if the modulator of the graph is given in the input.*

References

1. N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
2. M. Biró, M. Hujter, and Zs. Tuza. Precoloring extension. I. Interval graphs. *Discrete Math.*, 100(1-3):267–279, 1992.
3. B. Bollobás. On generalized graphs. *Acta Math. Acad. Sci. Hungar*, 16:447–452, 1965.
4. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inform. Process. Lett.*, 58(4):171–176, 1996.
5. L. Cai. Parameterized complexity of vertex colouring. *Discrete Appl. Math.*, 127:415–429, 2003.
6. M. C. Golumbic. *Algorithmic graph theory and perfect graphs.* Academic Press, New York, 1980.
7. M. Hujter and Zs. Tuza. Precoloring extension. II. Graph classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62(1):1–11, 1993.
8. M. Hujter and Zs. Tuza. Precoloring extension. III. Classes of perfect graphs. *Combin. Probab. Comput.*, 5(1):35–56, 1996.
9. H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28(5):1906–1922, 1999.
10. T. Kloks. *Treewidth*, volume 842 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1994. Computations and approximations.
11. D. Marx. Precoloring extension on chordal graphs, 2004. Submitted.
12. D. Marx. Precoloring extension on unit interval graphs, 2004. Submitted.
13. B. Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems (Udine, 1982)*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985.
14. D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.
15. A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. In *Algorithms – ESA 2003, 11th Annual European Symposium*, volume 2832 of *Lecture Notes in Comput. Sci.*, pages 482–493. Springer, Berlin, 2003.